Forty-Sixth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
September 23-26, 2008

Stochastic Analysis of Power-Aware Scheduling

Adam Wierman
Computer Science Department
California Institute of Technology

Lachlan L.H. Andrew
Computer Science Department
California Institute of Technology

Ao Tang
School of ECE
Cornell University

***Abstract*—Energy consumption in a computer system can be reduced by dynamic speed scaling, which adapts the processing speed to the current load. This paper studies the optimal way to adjust speed to balance mean response time and mean energy consumption, when jobs arrive as a Poisson process and processor sharing scheduling is used. Both bounds and asymptotics for the optimal speeds are provided. Interestingly, a simple scheme that halts when the system is idle and uses a static rate while the system is busy provides nearly the same performance as the optimal dynamic speed scaling. However, dynamic speed scaling which allocates a higher speed when more jobs are present significantly improves robustness to bursty traffic and mis-estimation of workload parameters.**

I. Introduction

Two threats to the growth of the internet have their roots in power consumption. The most pressing is that Moore's law has increased the thermal density of electronics to such an extent that cooling is a major concern, and has halted the previously inexorable increase in clock speeds. The longer term threat is that the need to reduce fossil fuel consumption requires all aspects of society to conserve energy, while aggregate internet energy consumption is a significant and growing fraction of the energy consumption of developed countries [1]. As a result, all modern system designs must consider the tradeoff between energy use and other performance metrics.

Power can often be saved simply by running devices more slowly. Dynamic speed scaling, which selectively reduces the speed when the load is light, reduces energy consumption with minimal impact on performance. It is widely implemented in current processors, in the form of Intel's SpeedStep and AMD's PowerNow.

Much of the theory of dynamic speed scaling [2]–[6] considers worst-case bounds. While such worst-case results are important for temperature management [7], or when the energy to a specific computer is constrained [8], [9], global energy consumption is affected by the average case, rather than the worst case. Consequently, this paper studies the average performance in a stochastic setting.

In particular, this paper seeks to minimize a weighted sum of the mean response time and the energy use per job. This performance metric has been studied both theoretically [10]–[12] and in implementations [13].

Algorithms are known [12] for finding the speeds which optimize this objective in a stochastic M/M/1/FCFS setting. However, these are highly recursive, and provide little insight. The goal of this paper is to identify simple structural properties of the optimal solutions, and to use them to compare the gain of unconstrained speed scaling with that of an optimized static design. The paper makes three main contributions.

First, the paper provides bounds on the performance of dynamic speed scaling (Section IV-A). Surprisingly, these bounds show that even an idealized version of dynamic speed scaling improves performance only marginally compared to a simple scheme where the server uses a static speed when busy and runs at speed 0 when idle – at most a factor of 2 for typical parameters and often less (see Section V).

Second, the paper provides bounds and asymptotics for the speeds used by the optimal dynamic speed scaling scheme (Sections IV-B and IV-C). These results provide insight into how the speeds scale with the arriving load, the queue length, and the relative cost of energy.

Third, the paper illustrates through analytic results and numerical experiments that, though dynamic speed scaling provides limited performance gains, it dramatically improves robustness to mis-estimation of workload parameters and bursty traffic (Section VI).

Note that many proofs are omitted from this document; all proofs can be found in [14].

II. Model and notation

In order to study the performance of dynamic speed scaling, we focus on a simple model: an M/GI/1 PS queue with controllable service rates, dependent on the queue length. In this model, jobs arrive to the server as a Poisson process with rate λ, have intrinsic sizes with mean $1/\mu$, and depart at rate $s_n\mu$ when there are n jobs in the system. Under static schemes, the (constant) service rate is denoted by s. Define the "load" as $\rho = \lambda/\mu$, and note that the ρ is not the fraction of time the server is busy.

The performance metric we consider is $\mathbb{E}[T] + \mathbb{E}[E]/\beta'$, where T is the response time of a job, E is the energy expended on a job, and β' represents how delay-averse the design is. It is often convenient to work with the expected cost per unit time, instead of per job, which by Little's law can be written as $z = \mathbb{E}[N] + \lambda\mathbb{E}[f(s)]/\beta'$, where N is the number of jobs in the system and $f(s)$ determines the power used when running at speed s.

The remaining piece of the model is to define the form of $f(s)$. The dynamic power of a circuit is typically a low-order polynomial in the speed [15]. As a result, we will model the power used by running at speed s by

$$\lambda\frac{f(s)}{\beta'} = \frac{s^\alpha}{\beta} \tag{1}$$

978-1-4244-2926-4/08/$25.00 ©2008 IEEE

where $\alpha > 1$ and β takes the role of β', but has dimension (time)$^{-\alpha}$. The cost per unit time then becomes

$$z = \mathbb{E}[N] + \frac{s^\alpha}{\beta}. \tag{2}$$

We will often focus on the case of $\alpha = 2$ to provide intuition.

The impact of the workload parameters ρ, β, and α can often be captured by $\gamma = \rho/\beta^{1/\alpha}$, which is a dimensionless measure. Also, it will often be convenient to use the a natural dimensionless unit of speed $s/\beta^{1/\alpha}$.

III. Power-aware speed selection

This paper considers two natural forms of speed scaling:

(i) *Gated static speed:* The server "gates" its clock (setting $s = 0$) if no jobs are present, and if jobs are present it works at a constant rate chosen to balance energy usage and response time.

(ii) *Dynamic speed scaling:* The server adapts its speed to the current number of requests present in the system.

The goal of this paper is to understand how to choose optimal speeds in each of these scenarios and to contrast the relative merits of each scheme. Clearly the expected cost is reduced each time the server is allowed to adjust its speed more dynamically. This must be traded against the costs of switching, such as a delay of up to tens of microseconds to change speeds [16]. The important question is "What is the magnitude of improvement at each level?" For our comparison, we will use idealized versions of each scheme. In particular, in each case we will assume that the server can be run at any desired speed in $[0, \infty)$ and ignore switching costs.

In this section, we will derive expressions for the optimal speeds in case (i). For case (ii), we will describe a numerical approach for calculating the optimal speeds which is due to George and Harrison [12]. Though this numerical approach is efficient, it provides little insight into the structure of the dynamic speeds or the overall performance. Providing such results will be the focus of Section IV.

A. The optimal static speed for a gated system

In the simplest dynamic speed scaling, a server either runs at a constant rate, or has its clock gated using zero dynamic power when the system is empty. We call this policy the "gated-static" policy, and denote its cost z_{gs}.

Since the server can gate its clock, the energy cost is only incurred ρ/s of the time, when the server is busy. Thus

$$z = \frac{\rho}{s - \rho} + \rho\frac{s^{\alpha-1}}{\beta}.$$

The optimum occurs when $s > \rho$ and

$$(\alpha - 1)s^{\alpha-2}(s - \rho)^2 = \beta. \tag{3}$$

When $\alpha = 2$, $s_{gs} = \rho + \sqrt{\beta}$. In general, define

$$G(\gamma; \alpha) = \sigma \quad \text{s.t. } \sigma > \gamma \\ (\alpha - 1)\sigma^\alpha(1 - \gamma/\sigma)^2 = 1. \tag{4}$$

The "gated-static" speed is $s_{gs} = \beta^{1/\alpha}G(\gamma; \alpha)$.

The following lemma bounds G.

Lemma 1. *For $\alpha \geq 2$,*

$$\gamma + \sqrt{\frac{\gamma^{2-\alpha}}{\alpha - 1}} \leq G(\gamma; \alpha) \leq (\alpha - 1)^{-1/\alpha} + \frac{2}{\alpha}\gamma \tag{5}$$

and the inequalities are reversed for $\alpha \leq 2$.

B. Optimal dynamic speed scaling

A popular alternative to static power management is to allow the speed to adjust dynamically to the number of requests in the system. The task of designing an optimal dynamic speed scaling scheme in our model can be viewed as a stochastic control problem.

We start the analysis by noting that we can simplify the problem dramatically with the following observation. An M/GI/1 PS system is well-known to be insensitive to the job size distribution. This still holds when the service rate is queue-length dependent since the policy still falls into the class of *symmetric policies* introduced by Kelly [17]. As a result, the mean response time and entire queue length distribution are affected by the service distribution through only its mean. Thus, we can consider an M/M/1 PS system. Further, the mean response time and entire queue length distribution are equivalent under all non-size based service distributions in the M/M/1 queue [17]. Thus, to determine the optimal dynamic speed scaling scheme for an M/GI/1 PS queue we need only consider an M/M/1 FCFS queue.

The "service rate control" problem in the M/M/1 FCFS queue has been studied extensively [12], [18], [19]. In particular, George and Harrison [12] provide an elegant solution to the problem of selecting the state-dependent processing speeds to minimize a weighted sum of an arbitrary "holding" cost with a "processing speed" cost. Specifically, the optimal state-dependent processing speeds can be framed as the solution to a stochastic dynamic program, to which [12] provides an efficient numerical solution. In the remainder of this section, we will provide an overview of this numerical approach. The core of this approach will form the basis of our derivation of bounds on the optimal speeds in Section IV.

We will describe the algorithm of [12] specialized to the case considered in this paper, where the holding cost in state n is simply n. Further, we will generalize the description to allow arbitrary arrival rates, λ. The solution starts with an estimate z of the minimal cost per unit time, including both the occupancy cost and the energy cost. As in [12], [19], [20], the minimum cost of returning from state n to the empty system is given by the dynamic program

$$v_n = \inf_{s \in A} \left\{ \frac{1}{\lambda + \mu s}\left[\lambda\frac{s^\alpha}{\beta} + n - z\right] \right. \\ \left. + \frac{\mu s}{\lambda + \mu s}v_{n-1} + \frac{\lambda}{\lambda + \mu s}v_{n+1} \right\}$$

where A is the set of available speeds. We will usually assume $A = \mathbb{R}^+ \cup \{0\}$. With the substitution $u_n = \lambda(v_n - v_{n-1})$, this can be written as [12], [20]

$$u_{n+1} = \sup_{s \in A} \left\{ z - n + \lambda\frac{s^\alpha}{\beta} + \frac{su_n}{\rho} \right\}. \tag{6}$$

Two additional functions are defined. First,

$$\phi(u) = \sup_{x \in A}\{ux/\rho - \lambda x^\alpha/\beta\} = (\alpha-1)\left(\frac{u}{\alpha\gamma}\right)^{\alpha/(\alpha-1)}. \quad (7)$$

Second, the minimum value of x which achieves this supremum, normalized to be dimensionless, is

$$\psi(u) = \frac{1}{\beta^{1/\alpha}} \min\left\{x : \frac{ux}{\rho} - \frac{\lambda x^\alpha}{\beta} = \phi(u)\right\} = \left(\frac{u}{\alpha\gamma}\right)^{1/(\alpha-1)}. \quad (8)$$

Given the estimate of z, u_n satisfy

$$u_1 = z \quad (9a)$$

$$u_{n+1} = \phi(u_n) - n + z. \quad (9b)$$

The optimal value of z can be found as the minimum value such that $(u_n)_{n=1}^\infty$ is an increasing sequence. This allows z to be found by an efficient binary search, after which u_n can in principle be found recursively.

The optimal speed in state n is then given by

$$\frac{s_n^*}{\beta^{1/\alpha}} = \psi(u_n). \quad (10)$$

This highlights the fact that $\gamma = \rho/\beta^{1/\alpha}$ provides the appropriate scaling of the workload information because the cost z, normalized speed $s\beta^{-1/\alpha}$ and variables u_n depend on λ, μ and β only through γ.

IV. Bounds on optimal dynamic speed scaling

In the prior section, we presented the optimal designs for gated-static and dynamic speed scaling. In the first case, the optimal speed was presented more-or-less explicitly, however in the third case we presented only a recursive numerical algorithm for determining the optimal dynamic speed scaling. In this section, we provide results exhibiting the structure of the optimal dynamic speeds and the performance they achieve.

The main results of this section are summarized in Table I. The bounds on z for arbitrary α are essentially tight (i.e., agree to leading order) in the limits of small or large γ. Due to the complicated form of the general results, we illustrate the bounds for the specific case of $\alpha = 2$ to provide insight. In particular, it is easy to see the behavior of s_n and z as a function of γ and n in the case of $\alpha = 2$. This leads to interesting observations. For example, it illustrates a connection between the optimal stochastic policy and policies analyzed in the worst-case model. In particular, Bansal, Pruhs and Stein [11] showed that, when nothing is known about future arrivals, a policy that gives speeds of the form $s_n = (n/(\alpha-1))^{1/\alpha}$ is constant-competitive, i.e., in the worst case the total cost is within a constant of optimal. This matches the asymptotic behavior of the bounds for $\alpha = 2$ for large n. This behavior can also be observed for general α (Lemma 7 and Theorem 4).

A. Bounds on cost

We start the analysis by providing bounds on z in this subsection, and then using the bounds on z to bound s_n^* above and below (Sections IV-B and IV-C).

Recall that z_{gs} is the total cost under gated-static.

Theorem 2.

$$\max\left(\gamma^\alpha, \gamma\alpha(\alpha-1)^{(1/\alpha)-1}\right) \le z \le z_{gs} = \frac{\gamma}{G(\gamma;\alpha)-\gamma} + \gamma G(\gamma;\alpha)^{\alpha-1}$$

Proof: The optimal cost z is bounded above by the cost of the gated-static policy, which is simply

$$z_{gs} = \frac{\gamma}{G(\gamma;\alpha)-\gamma} + \gamma G(\gamma;\alpha)^{\alpha-1}. \quad (15)$$

Two lower bounds can be obtained as follows.

In order to maintain stability, the time-average speed must satisfy $\mathbb{E}[s] \ge \rho$. But $z > \mathbb{E}[s^\alpha]/\beta \ge (\mathbb{E}[s])^\alpha/\beta$ by Jensen's inequality and the convexity of $(\cdot)^\alpha$. Thus

$$z > \frac{\mathbb{E}[s^\alpha]}{\beta} \ge \frac{\rho^\alpha}{\beta} = \gamma^\alpha. \quad (16)$$

For small loads, this bound is quite loose. Another bound comes from considering the minimum cost of processing a single job of size X, with no waiting time or processor sharing. It is optimal to serve the job at a constant rate [2]. Thus

$$\frac{z}{\lambda} \ge \mathbb{E}_X\left[\min_s\left(\frac{X}{s} + \frac{s^\alpha}{\beta}\frac{X}{s}\right)\right].$$

The right hand side is minimized for $s = (\beta/(\alpha-1))^{1/\alpha}$ independent of X, giving $z \ge \rho\beta^{-1/\alpha}\alpha(\alpha-1)^{(1/\alpha)-1}$. Thus

$$z \ge \max\left(\gamma^\alpha, \gamma\alpha(\alpha-1)^{(1/\alpha)-1}\right). \quad (17)$$

■

The form of the bounds on z are complicated, so it is useful to look at the particular case of $\alpha = 2$.

Corollary 3. *For $\alpha = 2$, gated-static has cost within a factor of 2 of optimal. Specifically,*

$$\max(\gamma^2, 2\gamma) \le z \le z_{gs} = \gamma^2 + 2\gamma. \quad (18)$$

It is perhaps surprising that such an idealized version of dynamic speed scaling provides such a small magnitude of improvement over a simplistic policy such as gated-static. In fact, the bound of 2 is very loose when γ is large or small. Further, empirically, the maximum ratios for typical α are below 1.1 (see Figure 2). Thus there is little to be gained by dynamic scaling in terms of *mean cost*. However, Section VI shows that dynamic scaling dramatically improves robustness.

A second interesting observation about Corollary 3 is that the expected response time under these power aware schemes remains bounded as the arrival rate λ grows. Specifically, by (16),

$$\mathbb{E}[T] = \frac{z}{\lambda} - \frac{\mathbb{E}[s^2/\beta]}{\lambda} \le \frac{2}{\mu\sqrt{\beta}}.$$

This is a marked contrast to the standard M/GI/1 queue.

TABLE I
BOUNDS ON TOTAL COSTS AND SPEED AS A FUNCTION OF THE NUMBER $n \geq 1$ OF JOBS IN THE SYSTEM.

For any α,

$$\max\left(\gamma^\alpha,\ \gamma\alpha(\alpha-1)^{(1/\alpha)-1}\right) \leq z \leq \frac{\gamma}{G(\gamma;\alpha)-\gamma} + \gamma G(\gamma;\alpha)^{\alpha-1} \qquad \text{Theorem 2} \qquad (11)$$

$$\sigma_n \leq \frac{s_n^*}{\beta^{1/\alpha}} \leq \left(\frac{1}{\alpha}\min_{\sigma>0}\left(\frac{n+\sigma^\alpha-\gamma^\alpha}{(\sigma-\gamma)} + \frac{\gamma}{(\sigma-\gamma)^2}\right)\right)^{1/(\alpha-1)} \qquad \text{Theorems 8 and 4} \qquad (12)$$

where σ_n satisfies $\sigma_n^{\alpha-1}((\alpha-1)\sigma_n - \alpha\gamma) \geq n - (\gamma/(G(\gamma;\alpha)-\gamma) + \gamma G(\gamma;\alpha)^{\alpha-1}$

For $\alpha = 2$,

$$\max\left(\gamma^2, 2\gamma\right) \leq z \leq \gamma^2 + 2\gamma \qquad \text{Corollary 3} \qquad (13)$$

$$\gamma + \sqrt{n-2\gamma} \leq \frac{s_n^*}{\sqrt{\beta}} \leq \gamma + \sqrt{n} + \min\left(\frac{\gamma}{2n},\ \frac{3}{2}\left(\frac{\gamma}{4}\right)^{1/3}\right) \qquad \text{Corollaries 9 and 5} \qquad (14)$$

For $\alpha = 2$ and $n < 2\gamma$, a lower bound on s_n results from linear interpolation between $\max(\gamma/2, 1)$ at $n = 1$ and γ at $n = 2\gamma$.

B. Upper bounds on the optimal dynamic speeds

We now move to providing upper bounds on the optimal dynamic speed scaling scheme.

Theorem 4. *For all n and α,*

$$u_n \leq \gamma\frac{n+\sigma^\alpha-\gamma^\alpha}{\sigma-\gamma} + \frac{\gamma^2}{(\sigma-\gamma)^2} \qquad (19)$$

for all $\sigma > 0$, whence

$$\frac{s_n^*}{\beta^{1/\alpha}} \leq \left(\frac{1}{\alpha}\min_{\sigma>0}\left(\frac{n+\sigma^\alpha-\gamma^\alpha}{(\sigma-\gamma)} + \frac{\gamma}{(\sigma-\gamma)^2}\right)\right)^{1/(\alpha-1)}. \qquad (20)$$

In particular, for $\sigma = \gamma + n^{1/\alpha}$,

$$u_n \leq n^{(\alpha-1)/\alpha}\,\gamma\,\left(1+(1+\gamma)^\alpha\right) + \gamma^2 \qquad (21)$$

which is concave in n.

Proof: As explained in [20], (6) can be rewritten as

$$u_n = \rho\min_{s_n}\left[\frac{s_n^\alpha/\beta + n + u_{n+1} - z}{s_n}\right]. \qquad (22)$$

Unrolling the dynamic program (22) gives a joint minimization over all s_n

$$\begin{aligned} u_n &= \rho\min_{s_n}\frac{1}{s_n}\Bigg[s_n^\alpha/\beta + n - z \\ &\quad + \rho\min_{s_{n+1}}\frac{1}{s_{n+1}}\left[s_{n+1}^\alpha/\beta + (n+1) - z + u_{n+2}\right]\Bigg] \\ &= \min_{s_i, i\geq n}\sum_{i=n}^{\infty}\left(\prod_{j=n}^{i}\frac{\rho}{s_j}\right)\left(s_i^\alpha/\beta + i - z\right). \end{aligned} \qquad (23)$$

An upper bound can be found by taking any (possibly suboptimal) choice of s_{n+i} for $i \geq 1$, and bounding the optimal z. Taking $s_i = \sigma\beta^{1/\alpha} > 0$ for all $i \geq n$ gives

$$\begin{aligned} u_n &\leq \min_{\sigma>0}\frac{\gamma}{\sigma}\sum_{j=0}^{\infty}\left(\frac{\gamma}{\sigma}\right)^j\left(\sigma^\alpha + (n+j) - z\right) \\ &= \gamma\min_{\sigma>0}\left[\frac{n+\sigma^\alpha-z}{\sigma-\gamma} + \frac{\gamma}{(\sigma-\gamma)^2}\right]. \end{aligned}$$

Since $z \geq \gamma^\alpha$ from (17), equation (19) follows. With (10), this establishes (20).

For $n = 0$, (21) holds since $u_0 = 0$. Otherwise, it follows from the inequality $\sigma^\alpha = n(1+\gamma n^{-1/\alpha})^\alpha \leq n(1+\gamma)^\alpha$ and the fact that $n^{-2/\alpha} \leq 1$. ∎

By specializing to the case when $\alpha = 2$, we can provide some intuition for the upper bound on the speeds. Factoring the difference of squares in the first term of (19) yeilds one increasing term and two decreasing terms. Minimizing pairs of these terms gives the following upper bounds on u_n.

Corollary 5. *For $\alpha = 2$,*

$$\frac{s_n^*}{\beta^{1/\alpha}} \leq \sqrt{n} + \gamma + \min\left(\frac{\gamma}{2n}, \frac{3}{2}\left(\frac{\gamma}{4}\right)^{1/3}\right). \qquad (24)$$

C. Lower bounds on the optimal dynamic speeds

Finally, we prove lower bounds on the dynamic speed scaling scheme. We begin by bounding the speed used when there is one job in the system. The following result is an immediate consequence of Corollary 3 and (9a).

Corollary 6. *For $\alpha = 2$,*

$$\max\left(\frac{\gamma}{2}, 1\right) \leq \frac{s_1^*}{\sqrt{\beta}} \leq \frac{\gamma}{2} + 1. \qquad (25)$$

Observe that the bounds in (25), like those in Corollary 3, are essentially tight for both large and small γ, but loose for γ near 1, especially the lower bound.

In conjunction with (21) and (10), the following lemma shows that speeds chosen to perform well in the worst-case are asymptotically optimal (for large n) in the stochastic model.

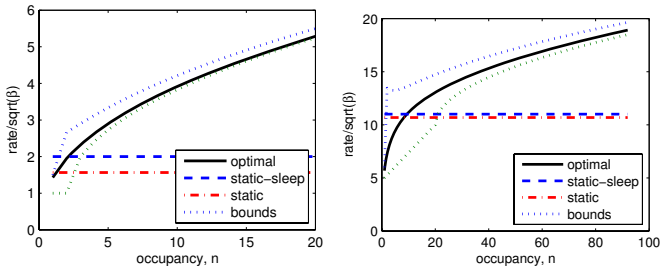
Lemma 7. *For sufficiently large n,*

$$\frac{s_n^*}{\beta^{1/\alpha}} > \left(\frac{n}{\alpha-1}\right)^{1/\alpha}. \qquad (26)$$

The following tighter bound on the optimal speeds is obtained by using $u_n \leq u_{n+1}$ and (15).

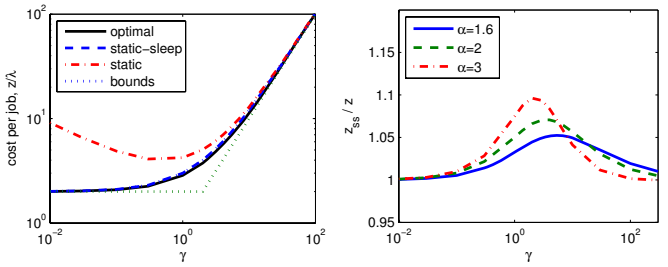
Theorem 8. *The scaled speed $\sigma_n = s_n^*/\beta^{1/\alpha}$ satisfies*

$$\sigma_n^{\alpha-1}\left((\alpha-1)\sigma_n - \alpha\gamma\right) \geq n - \frac{\gamma}{G(\gamma;\alpha)-\gamma} - \gamma G(\gamma;\alpha)^{\alpha-1}.$$



(a) $\gamma = 1$ (b) $\gamma = 10$

Fig. 1. Rate vs n, for $\alpha = 2$ and different energy-aware-load, γ.



(a) Absolute costs, $\alpha = 2$ (b) Ratio of cost for gated-static to optimal, z_{gs}/z.

Fig. 2. Cost z vs energy-aware-load γ.

For $\alpha = 2$, this becomes:

Corollary 9. *For $\alpha = 2$ and any $n \geq 2\gamma$,*

$$\frac{s_n^*}{\beta^{1/\alpha}} \geq \gamma + \sqrt{n - 2\gamma}. \tag{27}$$

This proves that the mode $n = \min_n\{s_n^* \geq \rho\}$ satisfies $n \leq 2\gamma$.

By the following lemma, linear interpolation between $\max(\gamma/2, 1)$ and γ gives a lower bound on s_n^* for $n < 2\gamma$.

Lemma 10. *The sequence u_n is strictly concave increasing.*

V. Comparing static and dynamic schemes

To this point, we have only provided analytic results. We now use numerical experiments to contrast static and dynamic schemes. In addition, these experiments will illustrate the tightness of the bounds proven in Section IV on the optimal dynamic speed scaling scheme.

We will start by contrasting the optimal speeds under each of the schemes. Figure 1 compares the optimal dynamic speeds with the optimal static speeds. Note that the bounds on the dynamic speeds are quite tight, especially when the number of jobs in the system, n, is large. For reference, the modes of the occupancy distributions are about 1 and 5, close to the points at which the optimal speed matches the static speeds. Note also that the optimal rate grows only slowly for n much larger than the typical occupancy. This is important since the range over which DVS is possible is limited [15].

Although the speed of the optimal scheme differs significantly from that of gated-static, the actual costs are very similar, as predicted by the remark after Corollary 3. This is shown in Figure 2. The bounds on the optimal speed are also very tight, both for large and small γ. Part (a) shows that the lower bound is loosest for intermediate γ, where the weights given to power and response time are comparable. Part (b) shows that gated-static (i.e., the upper bound) has very close to the optimal cost.

In addition to comparing the total cost of the schemes, it is important to contrast the mean response time and mean energy usage. Figure 3 shows the breakdown. A reference load of $\rho = 3$ with delay-aversion $\beta = 1$ and power scaling $\alpha = 2$ was compared against changing ρ for fixed γ, changing β for fixed ρ and changing α. Note $\gamma = 3$ was chosen to maximize the ratio of z_{gs}/z. The second scenario shows that when γ is held fixed, but the load ρ is reduced and delay-aversion is reduced commensurately, the energy consumption becomes negligible.

VI. Robust power-aware design

We have seen both analytically and numerically that (idealized) dynamic speed scaling only marginally reduces the cost compared to the simple gated-static. This raises the question of whether dynamic scaling is worth the complexity. This section illustrates one reason: *robustness*. Specifically, dynamic schemes provide significantly better performance in the face of bursty traffic and mis-estimation of workload.

We focus on robustness with respect to the load, ρ. The optimal speeds are sensitive to ρ, but in reality this parameter must be estimated, and will be time-varying.

It is easy to see the problems mis-estimation of ρ causes for static speed designs. If the load is not known, then the selected speed must be satisfactory for all possible anticipated loads. Consider the case that it is only known that $\rho \in [\underline{\rho}, \bar{\rho}]$. Let $z(\rho_1|\rho_2)$ denote the expected cost per unit time if the arrival rate is ρ_1, but the speed was optimized for ρ_2. Then, the robust design problem is to select the speed ρ' such that

$$\min_{\rho'} \max_{\rho \in [\underline{\rho}, \bar{\rho}]} z(\rho|\rho').$$

The optimal design is to provision for the highest foreseen load, i.e., $\max_{\rho \in [\underline{\rho}, \bar{\rho}]} z(\rho|\rho') = z(\bar{\rho}|\rho')$. However, this is wasteful in the typical case that the load is less than $\bar{\rho}$. The fragility of static speed designs is illustrated in Figure 4, which shows that when speed is underprovisioned, the server is unstable, and when it is overprovisioned the design is wasteful.

Optimal dynamic scaling is not immune to mis-estimation of ρ, since s_n^* is highly dependent on ρ. However, because the speed adapts to the queue length, dynamic scaling is more robust. Figure 4 shows this improvement.

Though the optimal dynamic scheme is more robust than a static scheme, robustness can be improved further. Specifically, consider the following speed scaling scheme that we term "linear". It scales the server speed in proportion to the queue length, i.e., $s_n/\beta^{1/\alpha} = n$. Figure 4 shows that the linear scaling provides significantly improved robustness when compared with the optimal dynamic scheme; indeed, the "optimal" scheme is only optimal for designs with $\rho \in [7, 14]$. Further, when ρ is in this region, the linear scaling provides only slightly higher cost than the optimal scaling. The price that linear scaling pays is that it requires very high processing speed

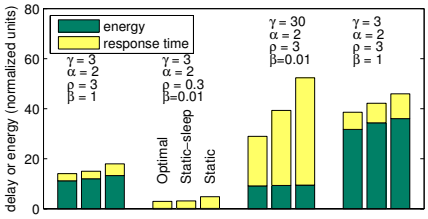


Fig. 3. Breakdown of $\mathbb{E}[T]$ and $\mathbb{E}[s^\alpha]$, for several scenarios.

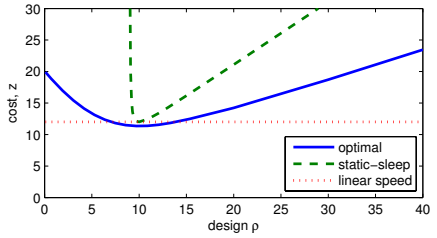


Fig. 4. Cost at load $\rho = 10$, when speeds are designed for "design ρ", using $\beta = 1$, $\alpha = 2$.

when the occupancy is high, which may not be supported by the hardware.

In addition to the numerical illustrations above, we can compare robustness analytically in the case of $\alpha = 2$. Theorem 11 shows that the cost of the linear scheme is exactly the same as the cost of the gated-static scheme when ρ is known exactly. Thus, the cost of the linear scheme is within a factor of 2 of optimal, even without using information about ρ.

Theorem 11. *When $\alpha = 2$, $z_{gs} = z_{lin}$. Thus, $z_{lin} \leq 2z$.*

Theorem 12. *Consider a system designed for target load ρ' that is operating at load ρ. When $\alpha = 2$,*

$$z_{lin} = \frac{\rho^2}{\beta} + 2\frac{\rho}{\sqrt{\beta}} \tag{28}$$

$$z_{ss} = z_{lin} + \frac{\rho}{\beta}\left(\frac{\epsilon^2}{\sqrt{\beta}+\epsilon}\right). \tag{29}$$

VII. Concluding Remarks

Speed scaling is an important method for reducing energy consumption in computer communication systems. Intrinsically, it trades off the mean response time and the mean energy consumption, and this paper provides insight into this tradeoff using a stochastic analysis.

Specifically, in the M/GI/1 PS model, both bounds and asymptotics for the optimal speed scaling scheme are provided. These bounds are tight for small and large γ and provide a number of insights, e.g., that the mean response time is bounded as the load grows under the optimal dynamic speed scaling and that the optimal dynamic speeds in the stochastic model match (for large n) dynamic speed scalings that have been shown to have good worst-case performance.

Surprisingly, the bounds also illustrate that a simple scheme which runs at speed 0 when the system is idle and uses a static rate while the system is busy provides performance within a factor of 2 of the optimal dynamic speed scaling. However, the value of dynamic speed scaling is also illustrated – dynamic speed scaling schemes provide significantly improved robustness to bursty traffic and mis-estimation of workload parameters. Interestingly, the dynamic scheme that optimizes the mean cost is no longer optimal when robustness is considered. In particular, a scheme that scales speeds linearly with n can provide significantly improved robustness while increasing cost only slightly.

VIII. Acknowledgement

This work was supported by grants from NSF CCF 0830511 and CNS 0435520, Microsoft Research and the Lee Center for Advanced Networking.

References

[1] J. Baliga, R. Ayre, W. Sorin, K. Hinton, and R. Tucker, "Energy consumption in access networks," in *IEEE Conf. Optical Fiber Communication (OFC)*, Feb. 2008, pp. 1–3.
[2] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced CPU energy," in *Proc. IEEE Symp. Foundations of Computer Science (FOCS)*, 1995, pp. 374–382.
[3] N. Bansal, T. Kimbrel, and K. Pruhs, "Speed scaling to manage energy and temperature," *J. ACM*, vol. 54, no. 1, pp. 1–39, Mar. 2007.
[4] Y. Zhu and F. Mueller, "Feedback EDF scheduling of real-time tasks exploiting dynamic voltage scaling," *Real Time Systems*, vol. 31, pp. 33–63, Dec. 2005.
[5] L. Yuan and G. Qu, "Analysis of energy reduction on dynamic voltage scaling-enabled systems," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 24, no. 12, pp. 1827–1837, Dec. 2005.
[6] S. Herbert and D. Marculescu, "Analysis of dynamic voltage/frequency scaling in chip-multiprocessors," in *Proc. ISLPED*, 2007, p. 6.
[7] S. Zhang and K. S. Catha, "Approximation algorithm for the temperature-aware scheduling problem," in *Proc. IEEE Int. Conf. Comp. Aided Design*, Nov. 2007, pp. 281–288.
[8] K. Pruhs, R. van Stee, and P. Uthaisombut, "Speed scaling of tasks with precedence constraints," in *Proc. Workshop on Approximation and Online Algorithms*, 2005.
[9] D. P. Bunde, "Power-aware scheduling for makespand and flow," in *Proc. ACM Symp. Parallel Alg. and Arch.*, 2006.
[10] S. Albers and H. Fujiwara, "Energy-efficient algorithms for flow time minimization," in *Lecture Notes in Computer Science (STACS)*, vol. 3884, 2006, pp. 621–633.
[11] N. Bansal, K. Pruhs, and C. Stein, "Speed scaling for weighted flow times," in *Proc. ACM-SIAM SODA*, 2007, pp. 805–813.
[12] J. M. George and J. M. Harrison, "Dynamic control of a queue with adjustable service rate," *Operations Research*, vol. 49, no. 5, pp. 720–731, Sep. 2001.
[13] J. O. Kephart, H. Chan, R. Das, D. W. Levine, G. Tesauro, F. Rawson, and C. Lefurgy, "Coordinating multiple autonomic managers to achieve specified power-performance tradeoffs," in *Proc. Int Conf Autonomic Computing*, 2007, p. 24.
[14] A. Wierman, L. L. H. Andrew, and A. Tang, "Power-aware speed scaling in processor sharing systems." [Online]. Available: http://www.cs.caltech.edu/ adamw/papers/power-2.pdf
[15] S. Kaxiras and M. Martonosi, *Computer Architecture Techniques for Power-Efficiency*. Morgan and Claypool, 2008.
[16] O. S. Unsal and I. Koren, "System-level power-aware deisgn techniques in real-time systems," *Proc. IEEE*, vol. 91, no. 7, pp. 1055–1069, 2003.
[17] F. P. Kelly, *Reversibility and Stochastic Networks*. Wiley, 1979.
[18] B. Ata and S. Shneorson, "Dynamic control of an M/M/1 service system with adjustable arrival and service rates," *Management Science*, vol. 51, no. 11, pp. 1778–1791, Nov. 2006.
[19] D. Low, "Optimal pricing policies for an M/M/s queue," *Operations Research*, vol. 22, pp. 545–561, 1974.
[20] J. Wijngaard and S. Stidham, "Forward recursion of Markov decision processes with skip-free-to-the-right transitions, part I: Theory and algorithm," *Mathematics of Operations Research*, vol. 11, no. 2, pp. 295–308, May 1986.